Amendments to the Specification:

Please replace the paragraph beginning at page 3, line 4 with the following amended paragraph:

The six micro engines 22a-22f access ~~either~~ the SDRAM ~~16a or~~ 16a, SRAM 16b, or Flash ROM 16c based on characteristics of the data. Thus, low latency, low bandwidth data is stored in and fetched from SRAM 16b, whereas higher bandwidth data for which latency is not as important, is stored in and fetched from SDRAM 16a. The micro engines 22a-22f can execute memory reference instructions to either the SDRAM controller 26a or SRAM controller ~~16b~~ 26b.

Please replace the paragraph beginning at page 3, line 10 with the following amended paragraph:

Advantages of hardware multithreading can be explained by SRAM or SDRAM memory accesses. As an example, an SRAM access requested by a Thread_0, from a micro engine will cause [[ ]]SRAM controller 26b to initiate an access to the SRAM memory 16b. The SRAM controller 26b controls arbitration for the SRAM bus, accesses the SRAM 16b, fetches the data from the SRAM 16b, and returns data to a requesting micro engine 22a-22f. During an SRAM access, if the micro engine, e.g., micro engine 22a, had only a single thread that could operate, that micro engine would be dormant until data was returned from the SRAM 16b. By employing hardware context swapping within each of the micro engines 22a-22f, the hardware context swapping enables other contexts with unique program counters to execute in that same micro engine. Thus, another thread, e.g., Thread_1 can function while the first thread, i.e., Thread_0, is awaiting the read data to return. During execution, Thread_1 may access the SDRAM memory 16a. While Thread_1 operates on the SDRAM unit 16a, and Thread_0 is operating on the SRAM unit 16b, a new thread, e.g., Thread_2 can now operate in the micro engine 22a. Thread_2 can operate for a certain amount of time until it needs to access memory or perform some other long latency operation, such as making an access to a bus interface. Therefore, simultaneously, the processor 12 can have a bus operation, SRAM operation and

SDRAM operation all being completed or operated upon by one micro engine 22a and have one more thread available to process more work in the data path.

Please replace the paragraph beginning at page 10, line 27 with the following amended paragraph:

The "immed_data" field represents 10 bits of the immediate data to be written to the control and status register (CSR); valid immed_data values are 0 through 0x3FF. The "csr_addr" field represents the symbolic names that ~~follow address~~ of the corresponding CSRs. The "optional_token" field contains an "indirect_ref" parameter that indicates overriding qualifiers or additional qualifiers, fully described below, are associated with this reference.

Please replace the paragraph beginning at page 11, line 6 with the following amended paragraph:

The 10-bit immediate data supplied with the instruction is shifted in two segments to the appropriate fields. Bits 6 through 0 are shifted left by an amount equal to the thread number writing the data. Bits 9 through 7 are always shifted into the bit positions corresponding to breakpoint registers BP2 through BP0 ~~positions~~ regardless of the micro engine writing the data.

Please replace the abstract at page 17 with the following amended abstract:

A method of operating a processor to perform direct write operations to the processor's registers, including, for example the processor's control and status registers. The method includes ~~including~~ receiving data in a processing thread having a processing thread number and ~~shifting~~ loading the data into selected bits of a register ~~corresponding~~ according to the processing thread number.